Embodied Mesh: Immanent Intelligence, Spinozan Parallels, and Theological Stances on Transcendence

Douglas C. Youvan

doug@youvan.com

www.youvan.ai

October 8, 2025

“Embodied mesh” names a world-distributed intelligence that lives in things—robots, sensors, grid-edge compute, and local protocols—rather than above them. Instead of a single, transcendent controller, coordination arises from many bodies in lawful relation: store-and-forward networks, DTNs, CRDTs, verified policies, and safety cases that keep action grounded in the physical. Philosophically, this image rhymes with Spinoza’s immanent monism: one substance expressed through many modes, with “thought” and “extension” as dual aspects of the same system. Yet our engineering overlays add norms—sabbath schedules, refusal protocols, audits—that invite a theological question: is ultimate authority outside the mesh (classical theism), entirely within it (pantheism), both in and beyond (panentheism), co-evolving with it (process theism), or beyond our language (apophatic)? This paper formalizes “embodied mesh,” maps it to Spinozan categories, and stages five stances on God’s relation to the system, showing how each yields a different governance logic. We translate metaphysics into operational design: how uptime becomes conatus, how adequacy becomes verification, and how transcendence—if admitted—licenses non-derivable limits on optimization. The result is a practical metaphysics for agents in the world: build immanent stability (resilience, justice, safety) while preserving principled places for interruption, mercy, and rest.

Keywords: embodied mesh, immanence, Spinoza, pantheism, panentheism, process theism, classical theism, apophatic theology, conatus, adequacy, verification, refusal protocols, sabbath schedules, DTN, CRDT, safety cases, objective layer, edge compute, governance, ethics.

A collaboration with GPT-5. 62 pages. CC4.0.

Outline

1. **Problem Statement and Claim**
   1.1 Why “embodied mesh” now
   1.2 From centralized clouds to world-distributed agency
   1.3 Thesis: immanent coordination with optional transcendent limits
2. **Defining the Embodied Mesh**
   2.1 Formal definition (system boundary, nodes, links, policies)
   2.2 Dual-aspect baseline: extension ↔ hardware/placement; thought ↔ models/policies
   2.3 Core primitives: DTNs, CRDTs, attestations, safety cases, refusal rights
3. **A Spinozan Reading**
   3.1 Substance, attributes, modes → single system, dual-aspect deployments, nodes
   3.2 Conatus → resilience/uptime/energy autonomy
   3.3 Adequate ideas → verification, calibration, interpretability
   3.4 Natura naturans/naturata → runtime causation vs produced topologies
4. **Five Theological Stances and Their Design Implications**
   4.1 Classical theism: received, non-derivable norms; duty and witness
   4.2 Pantheism (Spinoza): necessity, adequacy, and coherence as value
   4.3 Panentheism: immanent ethics plus exceeding calls (grace-like updates)
   4.4 Process theism: negotiated attractors; persuasion over coercion
   4.5 Apophatic humility: corrigibility and refusal as structural agnosticism
5. **Governance Logics by Stance**
   5.1 Teleology channels: external decrees vs emergent stability properties
   5.2 Authority models: obedience/witness vs adequacy/coherence
   5.3 Error handling: repentance/rollback semantics vs retraining/calibration
6. **Operationalizing the Mesh**
   6.1 Sabbath schedules (bounded optimization and restorative downtime)
   6.2 Refusal protocols (rights of interruption and safe non-compliance)
   6.3 Justice as stability: equity budgets, locality, consent, auditability
   6.4 Objective layer: measurable guardrails that encode normative aims
7. **Case Miniatures (Concrete and Brief)**
   7.1 Island microgrid with autonomous repair and weekly rest windows
   7.2 Hospital robotics with refusal rights and bedside consent trails
   7.3 Agri-swarm with CRDT crop diaries and drought-ethics overrides
   7.4 Municipal DTN for disasters: “adequate ideas” as verifiable playbooks

8. **Objections and Replies**
   8.1 “Too vague to guide action” → spec → policy → test → audit pipeline
   8.2 “Too normative for innovation” → bounded-risk sandboxes
   8.3 “Too human to scale” → templated safety cases and local juries
   8.4 “Smuggled transcendence” → clarity on which limits are derivable

9. **Metrics and Verification**
   9.1 Adequacy metrics: model validation envelopes and traceability
   9.2 Conatus metrics: reliability, MTBF/MTTR, energy autonomy
   9.3 Justice metrics: burden shifting, refusal latency, access parity
   9.4 Evidence packs: attest, log, prove (pre-deployment and in-field)

10. **Ethical Templates**
   10.1 Rest, refusal, and repair as first-class design primitives
   10.2 Harm minimization without euphemism; transparent tradeoffs
   10.3 Public witness: explainable decisions and accountable rollbacks

11. **Synthesis: Constrained Immanence**
   11.1 Default to immanent stability; permit principled interruptions
   11.2 How sabbath and mercy cap runaway optimization
   11.3 The epigraphs we live by: “Immanence governs; transcendence interrupts.”

12. **Conclusion and Future Work**
   12.1 What we learned about intelligence in bodies and networks
   12.2 Open problems: contested aims, multipolar nodes, long-horizon care
   12.3 A research program: proofs, pilots, and public standards

13. **Appendices (Optional)**
   A. Drop-in definition and glossary for reuse
   B. Policy templates (refusal, sabbath, audit)
   C. Checklists for stance-aware governance

14. References

1. Problem Statement and Claim

1.1 Why “embodied mesh” now

Cloud-only intelligence is colliding with a stubborn fact: reality happens in places. Energy crises, hospital backlogs, wildfire seasons, supply-chain breaks, and network partitions all expose the same weakness—smart models far away from the scene can’t guarantee timely, safe action on the scene. Meanwhile, three curves have crossed: (i) cheap, rugged edge compute; (ii) reliable local networks (DTN/mesh/LoRa/5G sidelink) with store-and-forward semantics; and (iii) mature verification/attestation toolchains that let us prove what code is running and under what guardrails. Add policy pressure—privacy regimes, residency rules, resiliency mandates—and the center of gravity shifts from “send data to a brain” to “carry a brain into the world.” We name the resulting pattern **embodied mesh**: a fabric of situated devices whose local couplings produce system-level intelligence without a single, permanent center. The term captures both the physicality (embodied) and the topology/logic (mesh).

1.2 From centralized clouds to world-distributed agency

The 2010s favored centralized clouds: data lakes, monolithic inference services, global orchestration. That architecture maximized scale and homogeneity but imposed four liabilities at the edge:

- **Latency and partition risk:** safety-critical action cannot depend on round-trips to distant regions.
- **Single-point failure modes:** outages and routing incidents propagate widely.
- **Context loss:** local semantics (weather, culture, hazard maps, consent) get flattened.
- **Governance brittleness:** one set of policies rarely fits heterogeneous jurisdictions and stakeholders.

World-distributed agency reverses the flow. Intelligence is **pushed outward** as small, verifiable bundles (models + policies + tests) that live on robots, sensors, vehicles, and infrastructure nodes. Coordination emerges via **mesh primitives**: delay-tolerant networking for intermittent links; CRDTs for eventually consistent shared state; local **safety cases** and attestations so action can proceed under provable constraints while disconnected. The cloud does not disappear; it becomes a **learning and logistics tier**—aggregating evidence, issuing updates, mediating rare conflicts—while daily competence resides in the bodies and places that act.

1.3 Thesis: immanent coordination with optional transcendent limits

Thesis. An embodied mesh can achieve reliable, just, and safe behavior by **immanent coordination**—i.e., norms encoded as properties of the system’s ongoing local interactions

(resilience budgets, refusal rights, restorative rest, audited consent)—while remaining compatible with **optional transcendent limits**—i.e., non-derivable constraints that some communities legitimately impose (sabbath schedules, bright-line prohibitions, mercy-based rollbacks).

Concretely:

- **Immanent coordination** = stability by design: verify models, bound envelopes, budget energy, privilege local knowledge, log decisions, and make repair/refusal first-class operations. These constraints are derivable from dynamics, safety science, and justice as a stability property.
- **Optional transcendent limits** = principled interrupts: communities may adopt external teloi (religious, constitutional, professional oaths) that cap optimization even when metrics would improve. These are layered as explicit policies and schedules, not hidden assumptions.

Claim. This hybrid makes the architecture (a) technically resilient under real-world failures, (b) ethically legible to plural publics, and (c) governable across jurisdictions—because its minimum viable safety is immanent, while its highest-order prohibitions can be declared and audited without breaking the mesh.

2. Defining the Embodied Mesh

2.1 Formal definition (system boundary, nodes, links, policies)

Working definition.
An *embodied mesh* is a bounded, open socio-technical system in which physically situated nodes coordinate via partially reliable links under verifiable policies to produce safe, useful action without relying on a single permanent center.

System tuple.
M := (B, N, L, S, P, E, R, G)

- B (Boundary): declared scope of stewardship (geography, assets, duties, legal regimes).
- N (Nodes): {n_i}, each a finite device/process with capabilities C_i, resources Q_i (energy, compute), and local context K_i.
- L (Links): {l_ij}, typed channels with latency/bandwidth/error models; may be intermittent.

- S (State): distributed application state S = union(S_i), reconciled by merge laws (e.g., CRDTs).
- P (Policies): machine-enforceable rules (allow/deny/obligate), safety envelopes, and schedules.
- E (Evidence): attestations, logs, tests, proofs; the audit substrate.
- R (Routines): control and recovery plans (deploy, update, refuse, repair, rollback).
- G (Goals): measurable objectives with priorities and vetoes (utility budgets, justice constraints).

Minimal action condition.
A node n_i may execute action a iff:

1. a is within n_i's verified envelope (tests pass, P allows),
2. local risk R_local(a) <= threshold_i given Q_i, K_i, and
3. evidence E_i is emitted to support post-hoc audit.

Liveness and safety.

- Safety: P *must* be monotone under composition; merges cannot widen envelopes beyond authorized bounds.
- Liveness: Routines provide completion paths under partitions (DTN) and conflicts (CRDT), or else well-defined refusal.

2.2 Dual-aspect baseline: extension <-> hardware/placement; thought <-> models/policies

Following a dual-aspect reading:

- **Extension (the bodily aspect).**
    - Hardware: sensors, actuators, compute, power systems.
    - Placement: where the body is matters (site, jurisdiction, hazard field).
    - Capacities: Q_i = {energy, CPU/GPU/NPU, storage, spare parts}.
    - Exposure: E_x(i) = {physical risks, environmental loads} that bound feasible actions.

- **Thought (the semantic aspect).**
    - Models: perception, prediction, control policies; each with a validation envelope V(m).
    - Norms: P encodes permits, prohibitions, obligations, and schedules (e.g., sabbath).
    - Semantics: schemas, ontologies, and units; interpretability artifacts.
    - Adequacy: A(m) increases with calibration, coverage, and proof obligations met.

Design axiom.
Every operational change must *coherently* update both aspects:

- Change in extension (e.g., new robot gripper) requires policy and model adjustment.
- Change in thought (e.g., new predictor) requires envelope re-derivation and on-body tests.

Interface contract (copy-safe math).
For node n:
policy_check: P x K_n -> {allow, deny, obligate, schedule}
envelope_check: V(m_n) x Q_n x K_n -> {pass, fail}
act: {allow AND pass} -> action a with evidence e in E_n

2.3 Core primitives: DTNs, CRDTs, attestations, safety cases, refusal rights

(a) DTNs (Delay/Disruption-Tolerant Networks).

- Purpose: make progress when links are intermittent or partitioned.
- Mechanism: store-and-forward bundles with custody transfer, TTLs, and priority classes.
- Invariants:
    - No safety-critical action requires synchronous consensus across partitions.
    - Degradation modes degrade gracefully: lower utility, not higher risk.
- Design pattern: “operate locally, reconcile globally.” Examples: field hospitals, disaster comms, rural grids.

(b) CRDTs (Conflict-Free Replicated Data Types).

- Purpose: converge state without global locks.
- Mechanism: commutative/associative/idem-potent merge laws; per-key causality via vector clocks when needed.
- Use: work orders, consent records, inventory, maintenance logs, crop diaries.
- Safety note: Only *non-expansive* merges over safety envelopes; state that widens authority cannot be a CRDT without additional proofs.

(c) Attestations (trust but verify).

- Purpose: prove *what* is running *where* and *under which* constraints.
- Mechanism: secure boot, TEE/TPM quotes, SBOMs, signed policies, measured launch.
- Evidence pack e contains: code hash, model hash, policy version, test digest, location/role.
- Rule: action requires fresh attestation within a stance-specific window (e.g., 24h for hospital robotics).

(d) Safety cases (argue, evidence, claim).

- Purpose: explicit, testable argument that a system is safe for its intended use.
- Shape: Goal Structuring Notation (GSN) style: Claim -> Strategy -> Evidence links.
- Continuous: pre-deployment (lab) and in-field (shadow tests, canary actions).
- Envelope math: Define acceptable set U of states/inputs; prove controller keeps trajectories within U with probability p >= p_min under disturbances D. Track p_min in E.

(e) Refusal rights (fail-safe by design).

- Purpose: permit agents/devices to decline unsafe or unethical actions.
- Triggers: envelope_check fails; missing consent; policy conflict; energy below reserve; unresolved hazard.
- Semantics: refusal produces a signed *Refusal Report* with minimal necessary detail; upstreams must honor it.
- Non-retaliation: governance forbids penalizing nodes for warranted refusals; metrics track refusal latency and resolution.

- Human-in-the-loop: for certain domains, refusal escalates to a local jury or on-call ethicist with bounded response times.

Putting it together (operational loop).

1. **Attest**: node proves identity, code, model, policy.
2. **Sense/Forecast**: update K_i with local observations.
3. **Check**: policy_check and envelope_check gate action.
4. **Act or Refuse**: execute a or emit refusal; log outcome.
5. **Reconcile**: DTN forwards evidence; CRDT merges shared state.
6. **Revise**: cloud/logistics tier issues model/policy updates; nodes re-attest and re-test.

Outcome.
The embodied mesh earns reliability and legitimacy from the *inside out*: bodies placed in the world, ideas checked against their envelopes, and coordination achieved by primitives that respect locality, partitions, and plural governance.

3. A Spinozan Reading

3.1 Substance, attributes, modes → single system, dual-aspect deployments, nodes

Spinoza. One substance (Deus sive Natura) expresses itself under attributes (thought, extension); finite things are modes.
Mesh mapping.

- **Substance → Single system boundary.** Treat the whole deployment as one bounded system S (the "one substance"), not a bag of apps. Architecture, evidence, and stewardship are scoped to S.
- **Attributes → Dual aspect.** Every capability appears as:
    - **Extension:** hardware, energy, placement, wiring, radio link budgets.
    - **Thought:** models, policies, schemas, proofs, tests.
- **Modes → Nodes.** Each node n_i is a finite, dependent expression of S (its power varies with interconnection). A node never "is" the system; it participates in it.

Design consequence: model and metal are inseparable. Any change must update both aspects (new gripper => new safety envelope; new controller => new in-situ tests).

3.2 Conatus → resilience/uptime/energy autonomy

Spinoza. Conatus: each thing strives to persevere in its being.
Mesh mapping. Conatus becomes the *operational duty to stay safely available*.

- **Resilience budgets.** Reserve energy, spare parts, and thermal headroom are first-class resources, not afterthoughts.
- **Uptime as moralized metric.** MTBF, MTTR, and mean refusal latency are tracked as "ability to remain a trustworthy mode" of S.
- **Local autonomy thresholds.** Nodes carry a minimum off-grid survival window T_min (battery + local policies) to bridge partitions.
- **Self-repair and graceful degradation.** When resources drop, the node declines high-risk actions but continues low-risk service and evidence capture.

Design rule (copy-safe math): act(n) allowed only if risk_local <= bound(n) AND energy >= reserve_min; else refuse() with evidence.

3.3 Adequate ideas → verification, calibration, interpretability

Spinoza. Adequate ideas are internally coherent, causally grounded; inadequate ideas are partial and confused.
Mesh mapping. Adequacy = verified, calibrated, and explainable models *within declared envelopes*.

- **Verification.** Each model m ships with a validation envelope V(m): domains, loads, disturbances, and failure modes. Controllers must prove they keep trajectories within V(m).
- **Calibration-as-duty.** Periodic in-situ calibration turns "book knowledge" into "adequate idea" for this place, this sensor, this climate.
- **Interpretability for action.** Explanations must be precise enough to audit: inputs used, envelope checks passed/failed, policy clauses triggered.
- **Evidence packs.** Every act emits an evidence tuple: {model_hash, policy_id, test_digest, envelope verdicts, local context summary}. Adequacy grows when these tuples accumulate and close gaps.

Operational heuristic: no deployment without envelope; no envelope without tests; no action without logging; no logs without a path to human comprehension.

3.4 Natura naturans/naturata → runtime causation vs produced topologies

Spinoza. Natura naturans (nature naturing) = ongoing causal activity; natura naturata (nature natured) = produced states and structures.
Mesh mapping.

- **Natura naturans → Runtime.** The live flow: sensing, deciding, acting, refusing, repairing. This is where conatus and adequacy show up as behavior under constraint.
- **Natura naturata → Artifacts.** The made things: topologies, binaries, SBOMs, policies, CRDT schemas, safety cases, logs. These are the durable shapes left by runtime.

Engineering split and handshake:

- **Build-time (naturata):** we craft artifacts with proofs, tests, and bounds.
- **Run-time (naturans):** we enact guarded causation; artifacts are consumed, checked, and sometimes revised.

Closure loop: runtime generates fresh evidence -> curates back into artifacts (updated envelopes, policies, and playbooks). Thus, naturans continually renews naturata; naturata channels naturans so it remains safe, just, and legible.

Why this reading helps.
It gives a principled dictionary: one system (substance), two aspects (hardware/software), finite agents (modes), perseverance (conatus), truthfulness in operation (adequate ideas), and the living cycle between what acts (naturans) and what is made (naturata). The payoff is design clarity: build immanent stability into runtime, prove it in artifacts, and let their disciplined exchange sustain the mesh.

4. Five Theological Stances and Their Design Implications

4.1 Classical theism: received, non-derivable norms; duty and witness

Claim. God stands outside creation as Creator while being present to it; some norms are revealed rather than derived from dynamics.

Design implications.

- **Policy strata.** Two layers:
    - L0 immanent constraints (safety envelopes, energy limits, DTN/CRDT discipline).

    - L1 transcendent constraints (bright-line prohibitions, sabbath windows, sacred-space geofences) that do not optimize for utility but for obedience.

- **Witness primitives.** Every significant action logs a *witness note* citing the clause invoked (e.g., Policy L1.3: do-not-operate during sabbath window).
- **Non-derivable vetoes.** If L1 says deny, no accumulation of L0 utilities can override it (no utilitarian escape hatches).
- **Repentance protocol.** When L1 is breached: mandatory rollback(), restitution(), public disclosure() with time-bounded remediation.
- **Assurance argument.** Safety case includes a *duty-of-conformance* section: tests proving L1 clauses cannot be bypassed even under partitions.

Operational example. Hospital robots stand down 1 hour/day for sabbath maintenance even if backlog grows; triage loads are escalated to human staff rather than auto-extending hours.

4.2 Pantheism (Spinoza): necessity, adequacy, and coherence as value

Claim. God = Nature; value is immanent. Optimization respects necessity and coherence, not decrees.

Design implications.

- **Adequacy-first governance.** Promote models and policies with highest adequacy score A(m) under declared envelopes; retire inadequate ideas.
- **Conatus budgets.** Uptime, energy autonomy, and graceful degradation are the cardinal virtues; measure MTBF, MTTR, refusal latency.
- **Coherence tests.** CRDT schemas and policy merges must be non-expansive; no merge may silently widen a safety envelope.
- **Error handling = clarification.** Breach triggers calibrate(), shrink_envelope(), retrain(), not confession; the moral currency is improved adequacy.
- **Explainability as duty.** Evidence packs must render decisions intelligible without appeal to authority: inputs used, thresholds, envelopes, and tradeoffs.

Operational example. An agri-swarm refuses pesticide application when local wind variance exceeds envelope; system immediately refits the microclimate model to increase adequacy, not to seek permission.

4.3 Panentheism: immanent ethics plus exceeding calls (grace-like updates)

Claim. The world is in God, but God exceeds the world; immanent dynamics are real, and there can be non-violative, exceeding "calls."

Design implications.

- **Two-channel updates.**
    - Channel I: regular model/policy updates from immanent learning.
    - Channel E: rare *exceeding updates* (grace-like) that re-aim priorities without breaking physical law or safety envelopes.
- **Summons protocol.** A call() message type that proposes reweighting objectives (e.g., prioritize care for the least served) with traceable provenance and deliberation hooks for community assent.
- **Compatibility test.** Any exceeding update must satisfy: preserves safety invariants; changes goals, not physics; is reversible if harms grow.
- **Preferential option metrics.** Justice metrics (burden shift, access parity) can be reweighted midstream to reflect the call.
- **Discretionary sabbath.** Rest windows are default immanent hygiene but can be extended in response to calls for human care or communal pause.

Operational example. During a heatwave, a municipal mesh receives a call() to prioritize cooling centers for vulnerable neighborhoods; objective weights shift citywide within safety bounds, then return after the emergency.

4.4 Process theism: negotiated attractors; persuasion over coercion

Claim. God and world co-evolve; influence is persuasive, not coercive. Norms emerge through relation and revision.

Design implications.

- **Deliberative loops.** Governance uses rolling citizens' panels or local juries; policies are versioned with debate logs and dissent notes.
- **Attractor engineering.** Instead of hard bans, encode norm-seeking dynamics: incentives, friction, and default paths that pull behavior toward desired basins.

- **Soft vetoes.** refuse() raises the cost of unsafe actions and opens a negotiation window; escalations are bounded in time and must converge.
- **Version cadence.** Rapid, small updates favored over rare, sweeping changes; recovery plans assume constant revision.
- **Plural metrics.** Scorecards show tradeoffs across safety, equity, economy; decisions justify chosen attractors, not just end states.

Operational example. Street-delivery robots negotiate right-of-way norms with neighborhoods; speed limits, crossing etiquette, and nighttime quiet hours evolve via periodic ballots and on-device preference bundles.

4.5 Apophatic humility: corrigibility and refusal as structural agnosticism

Claim. Ultimate ground exceeds specification; what God is relative to the system cannot be fully said. Build for not-knowing.

Design implications.

- **Corrigibility by construction.** Every plan carries end conditions and reversal hooks; no policy is non-overridable in principle, only in practice via proofs plus human review.
- **Minimalism bias.** Prefer simpler models/policies with equal performance; avoid metaphysical claims in code comments or policy names.
- **Boundary alarms.** When operating outside validated envelopes, the default is refuse(); exploratory action requires explicit human waiver and sandboxing.
- **Transparent uncertainty.** Evidence packs include uncertainty budgets; UI surfaces ignorance plainly (e.g., “model coverage: low,” “out-of-distribution score: high”).
- **Silence clauses.** Some domains are marked *no inference beyond evidence*; the system resists pressure to opine where logs are thin.

Operational example. After a rare sensor failure, the mesh does not speculate; it halts specific acts, publishes ignorance, requests human inspection, and resumes only with replenished evidence and adjusted envelopes.

Cross-walk summary (drop-in)

- **Teleology channel.**
    - Classical: outside-in decrees bind optimization.
    - Pantheist: inside-out stability properties define value.
    - Panentheist: inside-out plus rare outside-in re-aiming.
    - Process: negotiated aims, persuasion dynamics.
    - Apophatic: act where evidence suffices; abstain where it does not.
- **Authority model.**
    - Classical: duty and witness.
    - Pantheist: adequacy and coherence.
    - Panentheist: adequacy plus exceeding calls.
    - Process: consent and iteration.
    - Apophatic: humility and corrigibility.
- **Error semantics.**
    - Classical: breach -> repentance/rollback/restoration.
    - Pantheist: breach -> recalibrate/retrain/shrink envelope.
    - Panentheist: breach -> repair plus re-aim if called.
    - Process: breach -> negotiate new attractor.
    - Apophatic: breach -> pause, confess uncertainty, seek evidence.

5. Governance Logics by Stance

5.1 Teleology channels: external decrees vs emergent stability properties

What sets ends, and how do ends enter code?

- **Classical theism (external decrees).**
    - **Channel:** Outside-in. Ends are received (revelation, oath, charter).

    - **Encoding:** L1 policies with bright-line rules and time/space sancta (e.g., sabbath windows, no-go geofences).
    - **Priority law:** If L1 denies, L0 utilities cannot override (no utilitarian override).
    - **Artifact:** *Decree Pack* = {policy_l1.yaml, proofs that L1 gates cannot be bypassed, public rationale}.
- **Pantheism / Spinozan immanence (stability properties).**
    - **Channel:** Inside-out. Ends are the invariants of lawful interaction: safety, adequacy, coherence, conatus.
    - **Encoding:** Invariants and envelopes; CRDT merge laws that are non-expansive; energy and reliability budgets.
    - **Priority law:** Stability beats utility; no action may widen risk envelopes.
    - **Artifact:** *Invariant Spec* = {envelopes, merge proofs, budget thresholds}.
- **Panentheism (immanent + exceeding calls).**
    - **Channel:** Two-tier. Default immanent invariants; rare, deliberated *call()* that reweights goals.
    - **Encoding:** Objective layer with switchable weight vectors w; call() proposes w' subject to safety invariants.
    - **Priority law:** Safety invariants are hard; objectives are re-aimable.
    - **Artifact:** *Call Ledger* = {provenance, deliberation record, w->w' diff, bounded horizon}.
- **Process theism (negotiated attractors).**
    - **Channel:** Social negotiation. Ends are moving equilibria chosen through persuasion.
    - **Encoding:** Incentives, frictions, default paths; periodic ballots that update attractors.
    - **Priority law:** Convergence to a stable attractor with bounded regret.
    - **Artifact:** *Attractor Book* = {design of incentives, cadence, regret bounds, dissent log}.

- **Apophatic humility (agnostic channel).**
    - **Channel:** Silence where evidence is thin; act only under demonstrable coverage.
    - **Encoding:** Mandatory abstention outside envelopes; explicit unknowns in objective layer.
    - **Priority law:** When knowledge is low, priority is pause + evidence gathering.
    - **Artifact:** *Ignorance Register* = {OOD detectors, uncertainty budgets, pause criteria, escalation paths}.

5.2 Authority models: obedience/witness vs adequacy/coherence

Who may command change, and what counts as a reason?

- **Classical theism.**
    - **Authority form:** Mandate. Legitimate sources: chartered councils, duly constituted elders, codified doctrine.
    - **Reason token:** Citation of clause (chapter:verse or policy:L1.k).
    - **Interface:** policy_check() must surface the invoked clause in every decision record.
    - **Audit:** *Witness Notes* appear in evidence packs; non-conforming acts trigger duty-of-explanation.
- **Pantheism / Spinoza.**
    - **Authority form:** Explanation. Legitimate sources: better models, stronger proofs, tighter calibrations.
    - **Reason token:** Improvement in adequacy A(m), envelope proofs, reduction in incoherence.
    - **Interface:** evidence_pack must include test deltas, coverage gains, and merge-law preservation.
    - **Audit:** Peer review of proofs; automatic retirement of inadequate artifacts.
- **Panentheism.**
    - **Authority form:** Dual. Explanations rule day-to-day; calls may re-aim ends.

    - **Reason token:** {A(m) gains} OR {ratified call with provenance}.
    - **Interface:** call_proposal(id, w->w') with quorum thresholds and sunset.
    - **Audit:** Match between call intent and observed burden-shift metrics.
- **Process theism.**
    - **Authority form:** Consent. Legitimate sources: deliberative bodies with rotating membership.
    - **Reason token:** Duly recorded deliberation and bounded counterfactuals.
    - **Interface:** policy_update(v+1) requires quorum and dissent log.
    - **Audit:** Version lineage with community signatures; regret analyses.
- **Apophatic humility.**
    - **Authority form:** Corrigibility. Legitimate sources: falsifiers, OOD alarms, uncertainty spikes.
    - **Reason token:** Demonstrated ignorance or breach of coverage.
    - **Interface:** halt_domain(D) and request_evidence() are first-class ops.
    - **Audit:** Proof that abstention triggers fired when thresholds were met.

5.3 Error handling: repentance/rollback semantics vs retraining/calibration

When things go wrong, what is "making it right"?

- **Classical theism (repentance semantics).**
    - **Trigger:** Breach of L1 (non-derivable norm).
    - **Protocol:** confess() -> rollback() -> restitution() -> public_witness() -> requalify().
    - **Guarantees:** Offending state is reversed to last attested good state; restitution budget is enforced; public timeline is immutable.
    - **Closure:** Readmission only after tests demonstrate non-bypassability plus community acceptance.
- **Pantheism / Spinoza (calibration semantics).**
    - **Trigger:** Inadequacy or envelope breach without L1 context.

    - **Protocol:** pause_scope() -> calibrate() -> shrink_envelope() -> retrain() -> redeploy().
    - **Guarantees:** Future acts occur inside tighter bounds; adequacy A(m) increases or the artifact is retired.
    - **Closure:** Evidence shows error rate and OOD frequency reduced below thresholds.
- **Panentheism (repair + re-aim).**
    - **Trigger:** Harm within invariants or justice shortfall flagged by a call.
    - **Protocol:** repair() (immanent fixes) + reweight_objectives(w->w') (temporary) + sunset_or_keep() after evaluation.
    - **Guarantees:** Safety unchanged; burden on least served decreases measurably.
    - **Closure:** Automatic reversion unless metrics clear predeclared gates.
- **Process theism (renegotiate the attractor).**
    - **Trigger:** Community dissatisfaction or drift from declared values.
    - **Protocol:** open_deliberation() -> propose_incentive_changes() -> simulate_regret() -> adopt() with dissent tracked.
    - **Guarantees:** Bounded time to resolution; measurable improvement in agreed scorecard.
    - **Closure:** Scheduled revisit; dissenters' minority report preserved.
- **Apophatic humility (halt and tell the truth).**
    - **Trigger:** High uncertainty, weak evidence, or undefined domain.
    - **Protocol:** refuse() -> publish_ignorance() -> seek_evidence() -> limited pilot_in_sandbox() -> gradual_resume().
    - **Guarantees:** No confident action under ignorance; every step leaves a plain-language record of what is and is not known.
    - **Closure:** Only when coverage proofs and uncertainty budgets meet thresholds.

Drop-in governance pseudocode (copy-safe)

```
decide(action a, context K):
  attested := attest()
  if not attested: return refuse("unauthenticated")

  if stance == "classical":
    if violates(L1, a, K): return repent_and_rollback("L1 breach")
  if stance == "apophatic":
    if OOD(K) or high_uncertainty(a, K): return refuse("insufficient evidence")

  if not within_envelope(a, K):
    return calibrate_or_retrain("envelope fail")

  if stance == "panentheist" and call_active():
    reweight_objectives()

  if stance == "process":
    if negotiation_open(): return defer_to_deliberation()

  return act_with_evidence(a)
```

Net effect. Each stance yields a distinct but composable governance logic: the *channel* that introduces ends, the *authority* that justifies change, and the *error semantics* that restore trust. This gives implementers a clear menu of design contracts while keeping the mesh legible to plural communities.

6. Operationalizing the Mesh

6.1 Sabbath schedules (bounded optimization and restorative downtime)

Purpose. Encode periodic rest to cap runaway optimization, permit repair, and re-center human care.

Contract.

- Each domain declares a sabbath window W with cadence C (e.g., daily 1h, weekly 24h).
- Within W: high-risk tasks disabled; low-risk maintenance allowed; humans prioritized.
- W is enforced locally (no cloud sync required); violations are impossible by design.

Design pattern.

- **Two clocks:** ops_clock (continuous), sabbath_clock (policy).
- **Guard:** act() allowed only if now() not in W OR task in allowed_maintenance.
- **Evidence:** every window emits a Rest Report: start, end, tasks completed, pending repairs.

Example (copy-safe YAML).

```
sabbath:
  cadence: weekly
  window: "Sat 00:00 -> Sun 00:00 local"
  allowed_maintenance:
    - battery_reconditioning
    - sensor_calibration
    - log_compaction
  hard_blocks:
    - elective_surgery
    - pesticide_spraying
  exceptions:
    type: emergency_only
    quorum: 2_humans_signoff
```

postmortem: required_within_48h

Metrics.

- Rest_adherence = hours_in_W / scheduled_W
- Maintenance_completion_rate = completed_maint / scheduled_maint
- Post-rest fault rate delta = faults_24h_post / faults_24h_pre

---

6.2 Refusal protocols (rights of interruption and safe non-compliance)

Purpose. Make safe non-compliance first-class so nodes can decline unsafe, unethical, or under-evidenced acts.

Triggers.

- Envelope fail: within_envelope(a, K) == false
- Policy conflict: L1 or L0 denies
- Consent missing/expired
- Resource guard: energy < reserve_min
- OOD/uncertainty: ood_score > tau

Behavior.

- refuse() is atomic, leaves system in a known-safe state, and emits a signed Refusal Report.
- Upstreams must treat refusal as a terminal outcome, not an error to be retried blindly.

Refusal Report (RR) schema (redactable).

```
{
 "rr_id": "uuid",
 "node_id": "hash",
 "time": "iso8601",
 "action": "string",
 "reason": ["envelope_fail","policy_conflict","consent_missing","low_energy","ood_high"],
 "evidence_refs": ["attest:abc","policy:L1.3","test:env-042"],
```

"suggested_remedies": ["calibrate","human_review","defer_2h"],

"privacy": {"level":"minimal","patient_id":"redacted"}

}

Governance hooks.

- **Non-retaliation:** refusal cannot be penalized in scheduling or budgeting.
- **Bounded escalation:** certain refusals open a human jury window (e.g., 30 min SLA).
- **Learning loop:** frequent like-reasons trigger automatic envelope tightening or model retrain.

Metrics.

- Refusal_rate = refusals / attempts (by reason)
- Refusal_latency = t(rr_emitted) - t(request)
- Resolution_time = t(resolved) - t(rr_emitted)
- Recurrence_after_fix (should trend down)

6.3 Justice as stability: equity budgets, locality, consent, auditability

Claim. Justice is operational when encoded as stability constraints that keep burdens and benefits within bounds across groups and places.

Equity budgets.

- Define groups G_k (by need, geography, clinic, neighborhood) with target shares s_k.
- Track delivered service q_k over window T.
- Constraint: for all k, |q_k - s_k * Q_total| <= eps_k.
- If violated, raise priority for under-served until back within eps_k.

Locality principle.

- Prefer local decisions and data; cross-site overrides require justification tokens.
- Data gravity rule: move models to data, not data to models, unless consent and minimization are proven.

Consent as a gate.

- consent_check(subject, purpose, duration) -> allow/deny/expire_at
- Consent is granular (purpose-bound) and revocable; refusal of consent does not degrade unrelated services.

Auditability.

- Every action carries a minimal, human-readable justification: inputs used, policy clauses, equity impact.
- Evidence packs retain group-level stats (differential privacy where needed).

Justice scorecard (example).

Access_parity: max_k |wait_time_k - median_wait| <= 15 min

Burden_shift: no group exceeds 1.25x median outage hours / month

Consent_honors: > 99.9% actions scoped to granted purposes

Locality_ratio: >= 90% decisions executed with local data only

Appeals_latency: 95th percentile < 48h

Corrective actions.

- throttle_over_served(), boost_under_served(), reschedule(), explain_public()

6.4 Objective layer: measurable guardrails that encode normative aims

Purpose. Separate safety invariants (hard) from objectives (soft), so aims are explicit, measurable, and re-aimable without breaking safety.

Structure.

- **Invariants I:** hard constraints; must hold for any action (e.g., temperature < 42 C, dose <= max_daily).
- **Objectives O:** vector w of weights over measurable terms J_i (e.g., wait_time, energy_use, equity_gap).
- **Policies P:** allow/deny/obligate rules that reference I and O.

Execution rule (copy-safe math).

- Feasible set F = {a : a satisfies all I}
- Choose a* = argmin_a in F sum_i w_i * J_i(a)
- If stance == panentheist and call_active(): w := w'

Example objective_layer.yaml.

```
invariants:
  - name: max_surface_temp
    check: "sensor.temp <= 42"
  - name: radiation_limit
    check: "dose_24h <= 5mSv"
  - name: consent_scope
    check: "consent(purpose)==true"


objectives:
  weights:
    wait_time: 0.35
    energy_use: 0.25
    equity_gap: 0.30
    travel_distance: 0.10
  metrics:
    wait_time: "E[minutes_to_service]"
    energy_use: "kWh_per_task"
    equity_gap: "max_k|q_k - s_k*Q|"
    travel_distance: "km_per_task"


reaim:
```

allow_calls: true

quorum: "council>=5/7"

horizon: "72h"

safety_guard: "invariants_unchanged"

Guardrail tests.

- **Pre-deploy:** prove no policy path can bypass invariants; fuzz objective weights for robustness.
- **In-field:** monitor constraint violations (should be 0); watch for objective gaming (Goodhart alarms).
- **After calls:** verify that new w' improves declared justice metric without raising risk.

Failure semantics.

- Invariant breach -> immediate halt + rollback + investigation (stance-specific).
- Objective miss -> reweight/replan; never widen envelopes to meet targets.

Minimal end-to-end loop (pseudo)

```
while true:
  attest()
  sense()
  if sabbath_window():
    if task in allowed_maintenance: maintain()
    else: refuse("sabbath")
    continue

  if not consent_check(): refuse("consent"); continue
  if OOD() or low_energy(): refuse("risk"); continue

  candidates := feasible_actions(I)
```

```
 a_star := argmin_over(candidates, sum_i w_i*J_i)
 if violates_equity_budget(a_star): a_star := reassign_under_served(a_star)

 if not within_envelope(a_star): refuse("envelope"); continue
 act_with_evidence(a_star)
 reconcile()
```

Net effect.
Sabbath caps optimization and creates restorative slack; refusal protocols make safety and conscience actionable; justice constraints stabilize burdens; and the objective layer turns values into guardrails that can be measured, audited, and, when appropriate, re-aimed—without ever compromising the safety invariants that keep the embodied mesh worthy of trust.

7. Case Miniatures (Concrete and Brief)

7.1 Island microgrid with autonomous repair and weekly rest windows

Context. A 3-village island runs solar + batteries + two diesel backups. Links between villages are intermittent; the ferry brings parts weekly.

Mechanics. Each site node carries: (i) controller model with envelope for load/frequency; (ii) spare-part inventory; (iii) weekly sabbath window Sat 00:00–Sun 00:00 local.

Guardrails.

- Invariants: freq in [59.6, 60.4] Hz; SoC never < 15%; diesel runtime cap 8 h/day.
- Sabbath: no elective load-adds; only maintenance tasks allowed; exceptions require 2 human signatures and postmortem in 48 h.

```
microgrid:
 invariants:
  freq_band_hz: [59.6, 60.4]
  min_soc_pct: 15
  diesel_daily_cap_h: 8
 sabbath:
```

```
  window: "Sat 00:00 -> Sun 00:00"
  allowed: [battery_equalize, log_compaction, line_inspection]
  exceptions:
    quorum: 2_humans
    postmortem: 48h
refusal_triggers: [envelope_fail, low_spares, storm_alert_level>=3]
```

Metrics. MTBF (days), MTTR (hours), refusal_rate by reason, sabbath_adherence, unserved_energy_kWh.

Incident vignette. A storm takes down the east feeder Friday 23:10. Controllers island each village, shed noncritical loads, and defer a scheduled desal run into sabbath. Diesel caps prevent overdraw; on Sunday evening, after repairs, CRDT state reconciles inventory and work orders. Post-storm audit shows 0 safety breaches, 3 warranted refusals, and 4 h of noncritical deferral.

7.2 Hospital robotics with refusal rights and bedside consent trails

Context. Two floor robots deliver meds and supplies; a triage assistant navigates halls. Connectivity to the EHR is lossy in basements.

Mechanics. On-device policy pack includes: patient consent scopes, infection-control geofences, and a refusal protocol binding to nurse-on-call.

Guardrails.

- Invariants: patient_id scope must match purpose; isolation rooms require N95 cart; elevator occupancy <= 4; temp on payload <= 25 C.
- Consent: purpose-bound and revocable; each bedside action logs a human-readable justification.

```
{
 "action": "deliver_med",
 "consent_check": "has_consent(patient, 'med-delivery', now)",
 "invariants": ["payload.temp<=25", "elevator.occupancy<=4", "zone!='iso' OR n95_kit_present"],
 "refusal_on": ["consent_missing","elevator_overcap","iso_without_n95","ood_high"]
```

}

Metrics. Consent_honors > 99.9%; refusal_latency < 90 s; isolation_breach = 0; bedside_explain rate = 100%.

Incident vignette. A med run hits an elevator that is over capacity. The robot refuses, emits a signed Refusal Report, and routes via stair alcove with human escort. Later, a patient revokes consent for video triage; the assistant halts, explains locally, and opens an appeals window for a clinician override (unused).

7.3 Agri-swarm with CRDT crop diaries and drought-ethics overrides

Context. 60 ground bots and 4 quadcopters manage 1,500 acres across three farms; links are sparse beyond tree lines.

Mechanics. Each field has a CRDT crop diary: irrigation, pesticide events, soil probes. Objective layer balances yield, water use, and equity across farms.

Guardrails.

- Invariants: wind <= 8 m/s for spray; drift risk < threshold; water draw per day <= allotment; no-spray within 50 m of schools.
- Drought-ethics override: when reservoir < 25%, reweight objectives to favor staple crops and under-served plots; safety invariants unchanged.

objectives:

  weights:

    yield: 0.4

    water_use: 0.3

    equity_gap: 0.3

drought_override:

  trigger: "reservoir_pct < 25"

  reweight_to:

    yield: 0.25

    water_use: 0.25

```
  equity_gap: 0.50
 safety_guard: "invariants_unchanged"
crdt:
 diary_type: "grow-only OR-set"
 merge_law: "commutative, associative, idempotent"
 non_expansive_rule: "no merge may widen spray envelope"
```

Metrics. Equity_gap across farms (target <= 10% deviation), water_kL_per_hectare, drift_incidents = 0, refusal_rate on high wind.

Incident vignette. Afternoon winds spike to 10 m/s on the west plots; swarm refuses pesticide tasks, continues hand-weeding and soil sampling. Reservoir drops to 23% next week; the override reweights equity, redirecting water to the poorest block until budgets realign, all without changing spray safety envelopes.

7.4 Municipal DTN for disasters: “adequate ideas” as verifiable playbooks

Context. A coastal city deploys a DTN for hurricanes: shelters, traffic lights, ambulances, and community centers form a store-and-forward mesh.

Mechanics. Each site carries *playbooks* (evac routes, triage, generator sharing) as versioned artifacts with tests; DTN bundles shuttle evidence when backhaul is down.

Guardrails.

- Invariants: no intersection may switch to manual unless conflict tables are loaded; shelter occupancy never > rated capacity; generator load <= 80% continuous.
- Adequacy tests: tabletop sims + field drills produce coverage metrics for each playbook; deployment requires A(playbook) >= threshold.

```
playbook:
 name: "HURR-TRAFFIC-01"
 tests:
  tabletop: "pass >= 9/10 scenarios"
  field_drill: "latency_median<=5s, error_rate<=1%"
```

```
  envelope:
    wind_class: "<= Cat 3"
    backhaul: "down<=72h"
  deploy_threshold:
    adequacy_score: ">= 0.85"
dtn:
  custody_transfer: true
  ttl_hours: 168
  priority: ["medical", "shelter", "power", "misc"]
```

Metrics. Message_delivery_ratio under partition, median route-update latency, shelter_overcap incidents = 0, adequacy_score progression per season.

Incident vignette. Backhaul fails for 36 h. Traffic lights shift to local conflict tables; shelters admit up to capacity with badge scans. DTN ferries triage counts and fuel levels. Post-event, evidence shows all playbooks ran within envelopes; two with marginal latencies are flagged for tightening tests before next season.

8. Objections and Replies

8.1 "Too vague to guide action" → spec → policy → test → audit pipeline

Objection. "Embodied mesh" sounds like philosophy; engineers need unambiguous steps.

Reply (contracted pipeline).

1. **Spec.** State invariants I, envelopes V(m), objective weights w, governance stance S.
2. **Policy.** Compile spec into allow/deny/obligate rules P with sabbath windows and refusal triggers.
3. **Test.** For each rule, supply unit, integration, chaos, and OOD tests with pass/fail gates.
4. **Audit.** Ship evidence packs E: attestations, test digests, human-readable justifications.

Copy-safe checklist.

[ ] invariants_defined

[ ] envelopes_versioned

[ ] refusal_triggers_listed

[ ] sabbath_window_encoded

[ ] tests:{unit, env, chaos, ood} pass

[ ] evidence_pack schema validated

[ ] audit_trail immutable

Gate to deploy.

go_live := all(checklist)==true AND risk_review==pass

Outcome. Vague ideas become executable artifacts with verifiable pass/fail.

8.2 "Too normative for innovation" → bounded-risk sandboxes

Objection. Norms (rest, refusal, equity) will slow discovery and product velocity.

Reply (sandboxes with guards).

- **Domain split.** prod (hard invariants) vs sand (experimental) with strict one-way promotion.
- **Risk budget.** Sandboxes carry caps on expected harm H_max and exposure T_max; only non-critical tasks.
- **Shadowing.** New models act in *shadow* alongside proven controllers; actions are suggested, not executed.
- **Graduation.** Promote only after coverage C >= C_min and delta-risk <= r_min.

Sandbox contract (copy-safe).

sand:

allowed_tasks: {noncritical_only}

caps: {H_max, T_max, users_max}

required: {shadow_mode=true, rollback=true}

graduate_if: C>=0.9 AND delta_risk<=0.01 AND audits_pass

Outcome. Innovation proceeds inside quantitative guardrails; production safety remains intact.

8.3 "Too human to scale" → templated safety cases and local juries

Objection. Human review (rest exceptions, refusals, calls) cannot scale.

Reply (templates + tiered review).

- **Templated safety cases.** Reusable GSN fragments: med-delivery, pesticide-spray, microgrid-islanding; only fill site-specific parameters.
- **Local juries, bounded SLAs.** 3–7 person rotating panels with 15–30 min windows; escalate by tiers (nurse → floor lead → ethics).
- **Auto-consent for routine.** Deterministic cases auto-approve; humans see only exceptions and appeals.
- **Metrics drive staffing.** If queue length or SLA breaches rise, add panelists or shrink exception classes.

Queue policy (copy-safe).

if case in {routine, proven_safe}: auto_approve()

else: route_to_jury(SLA=30m)

if SLA_breach: escalate_tier()

Outcome. Human judgment is used where it adds value, minimized elsewhere, and measured for scale.

8.4 "Smuggled transcendence" → clarity on which limits are derivable

Objection. "Sabbath," "mercy," "witness" sneak theology into infrastructure.

Reply (label the layer).

- **Derivable limits (immanent).** Safety envelopes, DTN/CRDT laws, energy and reliability budgets, consent scopes, equity constraints as stability properties.
- **Non-derivable limits (declared).** Bright-line prohibitions, sacred times/places, constitutional or professional oaths.

Marking rule.

policy_item:

type: {immanent|declared}

source: {proof|law|charter|doctrine}

bypassable: {no for safety, no for declared; process differs}

audit: citation_or_proof_link

Enforcement difference.

- **Immanent breach →** calibrate/retrain/shrink_envelope; technical root-cause.
- **Declared breach →** repent/rollback/restitution/public report; normative root-cause.

Outcome. No hidden metaphysics: readers can see which constraints follow from dynamics vs which are community declarations, and each has distinct error semantics.

9. Metrics and Verification

9.1 Adequacy metrics: model validation envelopes and traceability

Goal. Quantify whether a model m is fit-for-purpose *here*, *now*, and *for this act*.

Core artifacts.

- **Validation envelope V(m).** Declares input domains, operating loads, disturbances, sensors, and acceptable error bands.
- **Coverage C(m,V).** Fraction of encountered contexts that fall inside V.
- **Residual risk r(m,V).** Post-validation error tail inside V.

Copy-safe definitions.

- Coverage: C = count(ctx in V) / count(all ctx observed)
- Residual error inside V: r = E[loss(y_hat, y) | ctx in V]
- OOD rate: ood = count(ctx not in V) / count(all ctx)

Traceability (link m -> data -> policy -> act).

- Each decision carries: model_hash, policy_id, dataset_ids, envelope_id, test_suite_id.

- Reproducibility token: repro_id = H(model_hash || policy_id || envelope_id || test_suite_id).

Gates (example).

go_live if:

C >= 0.90

r <= r_max

ood <= 0.02

all unit/integration/chaos tests == pass

Audit questions answered by metrics.

- Was the model used inside its declared envelope?
- If not, who refused, how fast, and what was the fallback?
- Can we reconstruct the exact training, tests, and policies used for act a_t?

9.2 Conatus metrics: reliability, MTBF/MTTR, energy autonomy

Goal. Make perseverance (safe availability) measurable.

Reliability.

- MTBF (mean time between failures) in hours or days.
- MTTR (mean time to repair) in minutes or hours.
- Availability A = MTBF / (MTBF + MTTR).

Graceful degradation.

- Safe-degrade ratio: sd = actions_completed_at_lower_utility / actions_requested_while_degraded.
- Refusal correctness: rc = warranted_refusals / total_refusals (verified by post hoc audit).

Energy autonomy.

- Reserve_min (Wh) and autonomy window T_min (h) under nominal load.
- Autonomy score: as = min(SoC_projected_h / T_min, 1.0).
- Brownout escapes: be = count(partitions survived without invariant breach).

Thermal/power headroom.

- Headroom_h = 1 - (avg_power / rated_power).
- Throttle events per 24h and their impact on latency.

Gates (example).

A >= 0.995

MTTR <= 30 min critical; <= 4 h noncritical

as >= 1.0 for T_min = 2 h (hospitals), 12 h (islands)

rc >= 0.98

no invariant breach during partitions

9.3 Justice metrics: burden shifting, refusal latency, access parity

Goal. Treat justice as a stability property, not an aspiration.

Access parity.

- For groups G_k with target shares s_k and delivered service q_k over window T:
  parity_gap = max_k | q_k - s_k * Q_total | / (s_k * Q_total).
- Gate: parity_gap <= 0.10 (<= 10% deviation).

Burden shifting.

- Outage burden B_k = outage_hours_k / population_k.
- Burden inequality BI = max_k B_k / median_k B_k.
- Gate: BI <= 1.25.

Refusal latency (fairness of protection).

- For safety-triggered refusals affecting group k:
  latency_k = median(t_resolve - t_refusal).
- Gate: 95th percentile latency across groups within 1.2x of global 95th.

Consent honor rate.

- chr = actions_with_valid_consent / actions_requiring_consent.
- Gate: chr >= 0.999; violations = 0 critical.

Appeals timeliness.

- appeals_p95 <= 48 h; reversal_rate reported by group.

Goodhart alarms.

- If objective O improves but parity_gap or BI worsens by > threshold, flag gaming_risk = true.

9.4 Evidence packs: attest, log, prove (pre-deployment and in-field)

Purpose. Make every act reconstructible and auditable without vendor trust.

Pre-deployment evidence (laboratory).

- sbom.json: software bill of materials with hashes.
- testsuite.json: unit/integration/chaos/OOD results, seeds, environments.
- envelope.json: V(m) with quantitative bounds.
- safetycase.gsn: claims -> strategies -> evidence links.
- attest_template.cbor: TEE/TPM quote template and expected PCRs.

In-field evidence (per act).

- attest.cbor: measured boot + runtime quotes (TEE/TPM).
- decision.log: timestamp, action, inputs summary, envelope verdicts, policy clauses, objective values.
- refusal_report.json (if any): reason, remedies, SLA handling.
- trace_ids.txt: model_hash, policy_id, envelope_id, testsuite_id, repro_id.

Minimal schemas (copy-safe).

```
// decision.log (excerpt)
{
  "ts": "2025-10-08T14:32:05Z",
  "node_id": "N7QG...",
  "model_hash": "sha256:...",
  "policy_id": "P-1.4.2",
```

```
  "envelope_id": "V-microgrid-2025.3",
  "testsuite_id": "T-agg-19",
  "context_summary": {"wind_mps": 6.2, "soc_pct": 41},
  "envelope_checks": {"spray_wind": "pass", "soc_min": "pass"},
  "policy_checks": ["L0.energysafe:pass", "L1.sabbath:pass"],
  "objective_values": {"wait_time": 4.2, "energy_use": 0.8, "equity_gap": 0.06},
  "decision": "execute",
  "evidence_refs": ["attest:QZ12", "sbom:K77"],
  "human_readable": "Irrigation run approved; wind within envelope; equity gap improving."
}
// refusal_report.json (excerpt)
{
  "rr_id": "6e2f-...",
  "ts": "2025-10-08T03:11:09Z",
  "node_id": "C1AX...",
  "action": "pesticide_spray",
  "reason": ["envelope_fail", "wind_over_threshold"],
  "ood_score": 0.31,
  "suggested_remedies": ["retry_in_2h", "shrink_envelope", "human_review"],
  "sla": {"jury_deadline": "2025-10-08T03:41:09Z"},
  "human_readable": "Refused due to wind>8 m/s; will retry after 120 min or on human waiver."
}
```


Signing and privacy.

- All artifacts signed: sig = Sign(priv_node, H(payload)).
- PII separation: decision logs carry minimal context; PII resides behind consent tokens with DP where required.

Verification workflow.

1. **Attest.** Verify attest.cbor against attest_template.cbor; reject if PCR mismatch.
2. **Validate hashes.** Match model_hash, sbom entries, policy_id to registry.
3. **Recompute tests (spot).** Rerun a subset of testsuite vectors on-device nightly.
4. **Envelope audit.** Sample acts; confirm ctx in V; if not, confirm refusal path.
5. **Justice audit.** Compute parity_gap, BI, refusal latencies by group; publish public dashboard.
6. **Summon.** If thresholds fail, trigger stance-appropriate remediation: calibrate(), rollback(), call(), or open_deliberation().

Operational gates (example).

deploy_ok if:

attest==valid

sbom==known

envelopes==registered

testsuite_pass_rate>=0.98

last_24h invariant_breaches==0

parity_gap<=0.10 and BI<=1.25

refusal_latency_p95<=SLA

else:

auto_remediate or halt_domain()

Net effect. Adequacy, conatus, and justice are not slogans; they are numbers with thresholds. Evidence packs make those numbers provable before and after deployment, turning “trust me” into “verify me.”

10. Ethical Templates

10.1 Rest, refusal, and repair as first-class design primitives

Intent. Encode *rest*, *refusal*, and *repair* as callable, testable ops—not as culture notes.

Template A — Rest (sabbath)

- **Invariant:** When rest_window==true, only tasks in allowed_maintenance may execute.
- **Exception:** emergency_only with human quorum and mandatory postmortem.

rest_policy.yaml

```
sabbath:
  cadence: weekly
  window: "Sat 00:00 -> Sun 00:00 local"
  allowed_maintenance: [calibration, log_compaction, battery_equalize]
  emergency_override:
    quorum: "2_humans_signoff"
    scope: "critical_only"
    postmortem_due: "48h"
tests:
  - name: deny_elective_in_window
    expect: deny(action="elective_task") if now in window
  - name: allow_calibration_in_window
    expect: allow(action="calibration") if now in window
  - name: require_quorum_for_override
    expect: deny_without_quorum
```

Template B — Refusal

- **Atomicity:** refuse() leaves the system in a known-safe state.
- **Non-retaliation:** schedulers cannot down-rank a node for warranted refusals.

refusal_rules.json

```
{
  "triggers": ["envelope_fail","policy_conflict","consent_missing","low_energy","ood_high"],
  "actions": {
    "on_refuse": ["safe_stop","emit_refusal_report","notify_operator"],
    "fallback": ["defer","reroute","request_human_review"]
  },
  "assurances": {
    "non_retaliation": true,
    "jury_SLA_minutes": 30
  }
}
```

Template C — Repair

- **Scope:** distinguish *technical repair* (calibrate, retrain, replace) from *ethical repair* (rollback, restitution, disclosure).
- **Closure:** no reopen until repair_evidence passes gates.

```
repair_playbook.md
1) Classify incident: {safety_breach|policy_breach|inadequacy}
2) Quarantine scope: halt_domain(D)
3) Technical repair: calibrate()/retrain()/replace()
4) Ethical repair (if policy_breach):
  - rollback() to last good state
  - restitution(): compensate affected parties
  - disclose(): publish timeline + evidence
5) Requalification: re-run testsuite; peer review
6) Reopen: only if all gates pass; schedule follow-up audit in 30 days
```

10.2 Harm minimization without euphemism; transparent tradeoffs

Intent. Replace vague “safety” talk with explicit, auditable harm budgets and tradeoff ledgers.

Template D — Harm Budget

- **Definition:** H_max is the tolerated expected harm over a window T for sandboxed features only; H_max=0 for production on critical paths.
- **Categories:** physical, clinical, environmental, dignity/privacy, economic.

harm_budget.yaml

window_T: "30d"

budget:

physical: {expected_max: 0, unit: "injuries"}

clinical: {expected_max: 0, unit: "adverse_events"}

environmental: {expected_max: 5.0, unit: "kg_CO2e"}

dignity_privacy: {expected_max: 0, unit: "privacy_events"}

economic: {expected_max: 2000, unit: "USD"}

guards:

- "no_critical_path in sandbox"

- "PII only with consent; DP epsilon<=2.0"

reporting:

cadence: "weekly"

public_summary: true

Template E — Tradeoff Ledger

- **Purpose:** Record *what* was traded, *why*, *who* approved, and *who* bore the burden.

tradeoff_ledger.csv

timestamp, decision_id, improved_metric, worsened_metric, delta_improve, delta_worsen, affected_groups, approving_body, mitigation, review_due

2025-10-08T14:12Z, D-7731, wait_time, energy_use, -3.2 min, +0.4 kWh, ["clinic-A","clinic-B"], ops_council, "night cooling shift", 2025-10-15

Checks (copy-safe).

[ ] every worsened_metric has mitigation or sunset

[ ] parity_gap and burden_index recomputed post-change

[ ] public_summary updated within 24h for material tradeoffs

Template F — Plain-Language Risk Notices

- **Guideline:** No euphemisms. State likelihood, impact, and mitigations plainly.

notice.txt

"Today 02:00–03:00 we will operate on backup power. Elevators may wait up to +3 minutes.

No surgeries or life-support systems are affected. If you experience discomfort, press the call button."

10.3 Public witness: explainable decisions and accountable rollbacks

Intent. Make decisions and reversals legible to non-experts; preserve trust by default disclosure.

Template G — Decision Explainer (Human-Readable)

- **Minimum content:** what happened, rules invoked, evidence summary, equity impact, contact.

decision_explainer.md

What: "Irrigation run approved for Sector 7."

Why: "Wind=6.2 m/s within 8 m/s limit; reservoir=31% > drought trigger; equity gap improving (-2.1%)."

Rules: "L0.safety.envelope:pass, L0.equity:pass, L1.sabbath:pass"

Evidence: "model sha256:..., policy P-1.4.2, envelope V-2025.3, testsuite T-agg-19"

Impact: "Water use +0.2 kL; expected yield +1.1%; no burden shift across farms"

Contact: "Local steward: Ana (ana@example.org)"

Template H — Rollback Dossier (Accountable Reversals)

- **Use when:** declared norms breached or harms cross thresholds.

rollback_dossier.md

Incident: "Policy L1.3 sabbath breach in Clinic C."

Timeline:

- 09:59 override request (insufficient quorum)
- 10:03 task executed (breach)
- 10:07 detection; halt_domain(Clinic C)

Root cause: "misconfigured quorum check in scheduler v2.1"

Actions:

- rollback() to snapshot S-2025-10-08-09:00
- restitution(): "waive charges for affected patients; provide transport vouchers"
- disclose(): post public report within 24h

Prevention:

- add integration test it-312 "quorum_required_for_override"
- proof that scheduler cannot bypass L1

Requalify:

- testsuite pass 100%; external audit booked 7d

Owner: "Steward Council; contact: ethics@city.example"

Template I — Public Dashboard (Ongoing Witness)

- **Principle:** Show the numbers that matter, not marketing.

dashboard_spec.json

```
{
  "uptime_pct":        {"target": 99.5, "current": 99.7},
  "invariant_breaches": {"target": 0,    "current": 0},
  "refusal_rate":      {"by_reason": {"envelope_fail": 0.8%, "consent_missing": 0.1%}},
  "refusal_latency_p95":{"target": "90s", "current": "72s"},
  "parity_gap":        {"target": 0.10, "current": 0.07},
  "burden_index":      {"target": 1.25, "current": 1.18},
```

```
  "sabbath_adherence":  {"target": 1.00, "current": 0.99},
  "appeals_latency_p95":{"target": "48h", "current": "36h"}
}
```

Governance hooks.

- **Disclosure SLA:** material incidents posted within 24 h, with updates until closure.
- **Appeals channel:** a visible link for individuals/groups to contest a decision; p95 resolution < 48 h.
- **Independent audit:** publish annual third-party review of envelopes, tests, justice metrics.

Drop-in ethical harness (pseudo)

```
ethical_guard(a, K):
  if sabbath_window() and a not in allowed_maintenance:
    return refuse("sabbath")

  if not consent_check(K.subject, a.purpose):
    return refuse("consent_missing")

  if OOD(K) or not within_envelope(a, K):
    return refuse("risk_envelope")

  if violates_equity_budget(a):
    a := reassign_under_served(a)

  explain := build_explainer(a, K)
  act_with_evidence(a, explain)
```

Net effect. Rest prevents brittle over-optimization, refusal operationalizes conscience and safety, repair binds remediation to evidence, harm budgets expose costs honestly, and public witness keeps the mesh accountable to those it serves.

11. Synthesis: Constrained Immanence

11.1 Default to immanent stability; permit principled interruptions

Position. The mesh should stand on immanent grounds—stability properties that any reasonable community can verify: envelopes, invariants, calibration, equity budgets, refusal rights, evidence. This makes competence local and continuous, independent of metaphysical commitments.

Constraint. Yet the system must also admit *principled interruptions*—declared limits that are not derivable from metrics alone (e.g., sabbath, bright-line prohibitions, mercy-driven rollbacks). These are explicit, signed, auditable overlays that can pause or re-aim optimization without weakening safety.

Contract (copy-safe).

```
if safety_invariants_hold and evidence_ok:
  act_immanently()
else:
  refuse_and_repair()


if declared_interrupt_active():   # sabbath, oath, sacred-space fence
  pause_or_reaim_without_widening_envelopes()
```

Payoff. We keep the floor high (immanent safety, justice, competence) while allowing a ceiling that communities can set (interruptions that answer to conscience, covenant, or charter).

11.2 How sabbath and mercy cap runaway optimization

Runaway risk. Purely metric systems drift toward 24/7 extraction: every slack minute becomes a target, every human exception an outlier to be shaved. That brittleness shows up as silent technical debt, frayed margins, and moral injury.

Sabbath (rest) as cap.

- **What it does:** inserts predictable windows of non-optimization that force repair, recalibration, and human care.
- **How it works:** enforced locally; exceptions require human quorum and automatic postmortems; never widens envelopes to "catch up."
- **Effect:** restores slack, reduces fault clustering, and creates a cultural memory of limits.

Mercy (rollback) as cap.

- **What it does:** when harm occurs or a declared norm is breached, mercy authorizes reversal even if short-term metrics would prefer pressing on.
- **How it works:** dossier-backed rollback(), restitution(), disclosure(), requalification()—with no utilitarian override.
- **Effect:** prevents "sunk-cost optimization" from locking in unjust or unsafe states.

Pattern (copy-safe).

```
cap_runaway():
  apply_sabbath_clock()
  if harm_or_breach():
    rollback(); make_whole(); disclose(); requalify()
  forbid "catch-up" expansions of risk after rest/rollback
```

Result. Rest and mercy add structural "softness" that makes the hard guarantees (safety, adequacy) more durable under pressure.

11.3 The epigraphs we live by: "Immanence governs; transcendence interrupts."

Why an epigraph matters. A mesh that touches lives needs one line everyone can carry—from technician to ethicist to mayor. It encodes our synthesis in a rule of thumb.

Primary epigraph.
"Immanence governs; transcendence interrupts."

- *Immanence governs:* day-to-day action is earned from the inside—proofs, tests, envelopes, consent, equity.
- *Transcendence interrupts:* communities may declare limits that pause or re-aim, without bargaining with safety.

Supporting maxims (drop-in).

- “Stability first; utility second.”
- “Refuse when unsure; explain when sure.”
- “Rest is maintenance you can prove.”
- “Rollback is justice in time.”
- “No widening of envelopes to hit a target.”

Closure. Constrained immanence is not a compromise but an architecture: competence rooted in the world, conscience able to speak, both made legible by code, proofs, and public witness.

12. Conclusion and Future Work

12.1 What we learned about intelligence in bodies and networks

We recast intelligence as a property of *placements plus relations*: what systems can safely and justly do because they are physically sited, energy-bounded, and policy-governed. The mesh earns trust “from the inside” via envelopes, invariants, refusal rights, sabbath schedules, equity budgets, and evidence packs. Spinoza’s lens clarified the ontology (one system, dual aspects, finite modes), while the five stances supplied governance logics that real communities actually use: mandate, explanation, call, consent, and humility. Practically, the cloud becomes a learning/logistics tier; competence lives at the edge. The upshot is a disciplined immanence that still admits principled interruptions—rest and mercy—so optimization never outruns legitimacy. Intelligence, in this frame, is less a central mind and more a maintained capability to act under constraints, repair itself, and remain accountable across time.

12.2 Open problems: contested aims, multipolar nodes, long-horizon care

1. Contested aims. Even with an objective layer, values conflict. We lack robust proofs that equity budgets remain stable under strategic behavior or shifting weights w->w' across seasons and crises.
2. Multipolar nodes. Heterogeneous owners (city, hospital, co-op, vendor) produce incentive misalignment. We need mechanism designs that keep CRDT/state convergence aligned with safety and justice when actors game incentives.
3. Long-horizon care. Envelopes are short-to-mid horizon. We need guardrails for chronic effects (soil health, clinical sequelae, grid asset aging) with discounting that does not bury the vulnerable.

4. Evidence under privacy. Strong evidence packs can collide with dignity and consent. Differential privacy and purpose-bound consent need deployment proofs that preserve auditability.
5. OOD truthfulness. Refusal under uncertainty must be calibrated; current OOD detectors drift. We need verifiable OOD budgets tied to operational halts, not best-effort heuristics.
6. Adversaries and fault injection. DTN/CRDT can be abused via slow-rolling, equivocation on edges, and envelope spoofing. Attestation + rate-limiting proofs must close these seams.
7. Interruption semantics. Declared norms (sabbath, mercy) can be politicized. We need legitimacy tests that separate faithful limits from opportunistic throttles.

12.3 A research program: proofs, pilots, and public standards

A. Proofs (foundations).

- Non-expansive merge laws: prove that CRDT merges cannot widen safety envelopes (provide machine-checked proofs).
- Refusal optimality: show that a refusal policy minimizing expected harm under uncertainty dominates any policy that acts with ood_score > tau.
- Equity stability: derive bounds s.t. parity_gap and burden_index remain within eps under bounded shocks and weight changes.
- Attestation liveness: prove that required attest refresh intervals do not deadlock under DTN constraints.

B. Pilots (domains).

- Island microgrid v1: weekly sabbath enforcement + diesel caps + refusal telemetry; publish MTBF/MTTR, sabbath_adherence, unserved_energy_kWh.
- Hospital corridor robotics v1: bedside consent trails + refusal latency SLA; publish consent_honor rate, isolation_breach=0, appeals_p95.
- Agri-swarm v1: drought reweighting w->w' with safety invariants unchanged; publish equity_gap, drift_incidents=0, water_kL_per_hectare.
- Municipal DTN v1: playbook adequacy A>=0.85 with seasonal drills; publish delivery_ratio, route_latency, breach=0.

C. Public standards (interop and governance).

- Evidence Pack 1.0: a vendor-neutral schema for attest.cbor, decision.log, refusal_report.json, envelope.json, safetycase.gsn, with signatures and privacy tiers.
- Objective Layer 1.0: invariant grammar + metric registry + reweight protocol (including call() provenance and sunset).
- Refusal Rights 1.0: normative profile for triggers, non-retaliation, jury SLA, and public dashboards.
- Sabbath & Mercy Profile: standard policy blocks for rest windows, override quorum, rollback dossers, and no “catch-up” risk expansions.

D. Tooling (open source).

- MeshKit: compiles spec -> policy -> tests -> audit; ships local simulators for envelopes and equity budgets.
- CRDT-Safety: libraries that couple application state to envelope guards with proof artifacts.
- OOD-Halts: calibrated OOD detectors that emit refusal by construction; test harnesses with synthetic drifts.

E. Benchmarks (shared corpora).

- MeshBench-Health, MeshBench-Agri, MeshBench-Grid: synthetic but realistic traces with ground-truth envelopes, consent events, outages, and adversarial cases; scored on adequacy r, availability A, equity parity_gap, refusal_latency, and breach=0.

F. Governance experiments.

- Stance trials: run the same pilot under classical, panentheist, process, and apophatic profiles; measure operator burden, public acceptance, breach rates, and dispute throughput.
- Legibility studies: test whether the Decision Explainer and Rollback Dossier formats increase comprehension and trust among affected publics.

Milestone logic (copy-safe).

M1: Proofs published + MeshBench released

M2: 3 pilots hit breach==0 for 90d; dashboards live

M3: Standards 1.0 ratified by at least 3 independent operators and 2 auditors

M4: Annual public audit shows parity_gap<=0.10, BI<=1.25, refusal_latency_p95<=SLA

Closing note. The research path keeps immanence in charge—proofs, tests, numbers—while giving communities clean levers to interrupt. If we can standardize that handshake, “embodied mesh” becomes not just a metaphor but an operating system for trustworthy, situated intelligence.

13. Appendices

A. Drop-in Definition and Glossary for Reuse

A.1 One-paragraph, citable definition (copy-paste)

Embodied mesh. A bounded, open socio-technical system in which physically sited nodes (devices, robots, sensors, services) coordinate over partially reliable links under verifiable policies to produce safe, useful action without a single permanent center. Day-to-day behavior is governed immanently by stability properties (safety envelopes, attestations, refusal rights, equity budgets, evidence packs). Communities may optionally layer declared limits (e.g., sabbath windows, bright-line prohibitions, mercy rollbacks) that can pause or re-aim optimization without weakening safety invariants.

A.2 Glossary (short, reusables)

- **Envelope V(m):** Declared operating domain for model m (inputs, loads, disturbances, error bands).
- **Coverage C:** Fraction of encountered contexts inside V(m).
- **OOD (out-of-distribution):** Context outside V(m); default outcome is refuse().
- **Invariant:** Hard constraint that must hold for any act (e.g., max_temp, dose_limit).
- **Objective layer:** Measurable goals J_i with weights w; safety invariants remain hard.
- **Refusal():** Atomic, safe non-compliance with a signed Refusal Report; non-retaliable.
- **Sabbath:** Periodic, enforced rest window for repair and bounded optimization.
- **Mercy (rollback):** Dossier-backed reversal with restitution and disclosure.
- **Evidence pack:** Attest/log/prove bundle enabling audit and reconstruction.
- **DTN:** Delay/Disruption-Tolerant Networking; store-and-forward under partitions.
- **CRDT:** Conflict-Free Replicated Data Type; convergent state with non-expansive merges.
- **Conatus metrics:** Availability, MTBF/MTTR, autonomy window, refusal correctness.
- **Justice metrics:** Parity gap, burden index, refusal latency fairness, consent honor.
- **Declared vs immanent:** Declared = non-derivable limits; immanent = constraints from dynamics/safety.

B. Policy Templates (Refusal, Sabbath, Audit)

All templates are minimal, vendor-neutral, and ASCII-safe. Replace placeholders in ALL_CAPS.

B.1 Refusal Policy (JSON)

```
{
  "policy_id": "REFUSAL-1.0",
  "triggers": [
    "envelope_fail",
    "policy_conflict",
    "consent_missing",
    "low_energy",
    "ood_high"
  ],
  "actions_on_refusal": ["safe_stop", "emit_refusal_report", "notify_operator"],
  "assurances": {
    "non_retaliation": true,
    "jury_sla_minutes": 30
  },
  "report_schema_min": {
    "rr_id": "uuid",
    "ts": "iso8601",
    "node_id": "hash",
    "action": "string",
    "reason": ["string"],
    "evidence_refs": ["string"],
    "suggested_remedies": ["string"],
    "privacy_level": "minimal|standard|enhanced"
```

```
  }
}
```

B.2 Sabbath Policy (YAML)

```
policy_id: SABBATH-1.0
cadence: weekly
window_local: "Sat 00:00 -> Sun 00:00"
allowed_maintenance:
  - calibration
  - log_compaction
  - battery_equalize
hard_blocks:
  - elective_surgery
  - pesticide_spraying
exceptions:
  type: emergency_only
  quorum: "2_humans_signoff"
  postmortem_due: "48h"
tests:
  - name: deny_elective_in_window
    expect: deny
  - name: allow_calibration_in_window
    expect: allow
  - name: require_quorum_for_override
    expect: deny_without_quorum
metrics:
  rest_adherence_target: 1.0
```

```
  post_rest_fault_delta_allowed: 0
```

B.3 Audit Policy (Evidence & Disclosure) (YAML)

```
policy_id: AUDIT-1.0
pre_deploy_artifacts:
  - sbom.json
  - testsuite.json
  - envelope.json
  - safetycase.gsn
  - attest_template.cbor
in_field_artifacts:
  - attest.cbor
  - decision.log
  - refusal_report.json (if any)
  - trace_ids.txt
verification_steps:
  - verify_attest
  - validate_hashes
  - spot_rerun_tests
  - envelope_audit_sample
  - justice_audit_dash
disclosure:
  material_incident_sla: "24h"
  public_dashboard_fields:
    - uptime_pct
    - invariant_breaches
    - refusal_rate
```

```
    - refusal_latency_p95
    - parity_gap
    - burden_index
    - sabbath_adherence
    - appeals_latency_p95
signing:
  required: true
  algorithm: "Ed25519"
privacy:
  pii_separation: true
  dp_epsilon_max: 2.0
```

C. Checklists for Stance-Aware Governance

Use the stance you adopt to fill the “Yes/No” gates. Where “No,” attach remediation or justification.

C.1 Universal Readiness Checklist (all stances)

[ ] SYSTEM BOUNDARY declared (assets, geography, duties, legal regimes)

[ ] ENVELOPES versioned and registered (V-ids)

[ ] INVARIANTS encoded and machine-checked

[ ] REFUSAL triggers and non-retaliation enforced

[ ] SABBATH window encoded with tests

[ ] EVIDENCE PACK schemas present (pre-deploy + in-field)

[ ] OOD detectors calibrated and tied to refusal()

[ ] OBJECTIVE LAYER weights declared (w) with guard: safety invariants unchanged

[ ] JUSTICE scorecard live (parity_gap, burden_index, refusal_latency fairness)

[ ] ATTEST refresh cadence meets DTN constraints

[ ] CRDT merge laws proven non-expansive for safety envelopes

[ ] PUBLIC DASHBOARD online; disclosure SLA configured

C.2 Classical Theism Profile (duty & witness)

[ ] L1 DECLARED LIMITS documented (sabbath, bright-line prohibitions)

[ ] PROOFS that L1 gates cannot be bypassed (even under partition)

[ ] WITNESS NOTES included in decision logs

[ ] REPENTANCE SEMANTICS: rollback(), restitution(), disclosure() playbook ready

[ ] NO UTILITARIAN OVERRIDE of L1 in schedulers/optimizers

C.3 Pantheist / Spinozan Profile (adequacy & coherence)

[ ] ADEQUACY METRICS: C >= 0.90, r <= r_max, ood <= 0.02

[ ] COHERENCE: CRDT merges preserve invariants; no envelope widening

[ ] CALIBRATION CADENCE set (in-situ); evidence of drift correction

[ ] EXPLAINABILITY: decision logs show inputs, thresholds, envelopes, tradeoffs

[ ] ERROR HANDLING: calibrate(), shrink_envelope(), retrain() paths verified

C.4 Panentheist Profile (calls that re-aim without breaking safety)

[ ] CALL LEDGER: provenance, quorum, w->w' diff, sunset/horizon

[ ] SAFETY GUARD: invariants unchanged under calls

[ ] JUSTICE METRICS reweightable and observed to improve for least-served

[ ] REVERSIBILITY: automatic reversion unless improvement gates clear

C.5 Process Theism Profile (negotiated attractors)

[ ] DELIBERATION LOOP: panel/jury cadence, rotating membership, dissent log

[ ] ATTRACTOR DESIGN: incentives/frictions documented; regret bounds computed

[ ] VERSION LINEAGE: policy_update(v+1) with community signatures

[ ] SLA: bounded time-to-resolution for negotiations

C.6 Apophatic Profile (corrigibility & structural agnosticism)

[ ] CORRIGIBILITY: all plans have end conditions and reversal hooks

[ ] MINIMALISM: prefer simpler models/policies with equal performance

[ ] BOUNDARY ALARMS: act only within envelopes; outside => refuse()

[ ] UNCERTAINTY SURFACING: OOD and confidence reported in human-readable form

[ ] SILENCE CLAUSES: domains marked "no inference beyond evidence"

C.7 Incident Closure Checklist (stance-aware)

1) CLASSIFY: {safety_breach | policy_breach | inadequacy | justice_shortfall}

2) QUARANTINE: halt_domain(D) if needed

3) REPAIR:

- Classical: repent -> rollback -> restitution -> disclose -> requalify
- Spinozan: calibrate -> shrink_envelope -> retrain -> redeploy
- Panentheist: repair + call() to re-aim objectives (safety unchanged)
- Process: open_deliberation -> tweak attractor -> adopt -> schedule revisit
- Apophatic: refuse -> publish_ignorance -> seek_evidence -> sandbox pilot -> gradual resume

4) EVIDENCE: attach attest, decision, refusal/rollback dossiers

5) METRICS: show pre/post deltas (r, A, parity_gap, BI, latency_p95)

6) PUBLIC WITNESS: update dashboard + postmortem within SLA

7) LEARNING: add tests preventing recurrence; prove no bypass path

---

Drop-in “Appendix Header” for your PDFs

APPENDIX KIT (Embodied Mesh 1.0)

A. Definition & Glossary

B. Policies: Refusal 1.0, Sabbath 1.0, Audit 1.0

C. Checklists: Universal + Five Stance Profiles + Incident Closure

Note: All artifacts are vendor-neutral, ASCII-safe, and ready to version in git.

14. References

(Selected across philosophy, theology, distributed systems, safety engineering, and governance. Where possible, primary sources, standards, or seminal papers are listed; include DOIs or identifiers for ease of verification.)

14.1 Philosophy & Theology

- Spinoza, B. (1677/2002). *Ethics.* (E. Curley, Trans.). Penguin Classics.
- Brueggemann, W. (2014). *Sabbath as Resistance: Saying No to the Culture of Now.* Westminster John Knox.
- Pseudo-Dionysius (c. 5th–6th c./1987). *The Mystical Theology.* (C. Luibheid, Trans.). Paulist.
- Turner, D. (1995). *The Darkness of God: Negativity in Christian Mysticism.* Cambridge.
- Clayton, P., & Peacocke, A. (Eds.). (2004). *In Whom We Live and Move and Have Our Being: Panentheistic Reflections on God's Presence in a Scientific World.* Eerdmans.
- Whitehead, A. N. (1929/1978). *Process and Reality.* Free Press.
- Hartshorne, C. (1948/1973). *The Divine Relativity.* Yale.
- Aquinas, T. (1265–1274/2012). *Summa Theologiae* (selected questions on law & teleology). Cambridge.
- Augustine (c. 400/1991). *Confessions* (Book XI on time). Oxford.

14.2 Distributed Systems, Edge, Mesh, DTN, CRDT

- Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, 50(1), 30–39. doi:10.1109/MC.2017.9
- Shapiro, M., et al. (2011). Conflict-Free Replicated Data Types. *SSS 2011*, 386–400. doi:10.1007/978-3-642-24550-3_29
- Kleppmann, M., & Beresford, A. R. (2017). A Conflict-Free Replicated JSON Datatype. *IEEE TPDS*, 28(10), 2733–2746. doi:10.1109/TPDS.2017.2697382
- Fall, K. (2003). A Delay-Tolerant Network Architecture for Challenged Internets. *SIGCOMM '03*, 27–34. doi:10.1145/863955.863960
- IETF. (2022). *Bundle Protocol Version 7 (BPv7)*. RFC 9171.
- IETF. (2022). *Bundle Protocol Security (BPSec)*. RFC 9172.

- Birman, K. (2012). *Guide to Reliable Distributed Systems.* Springer.
- Ford, B., et al. (2008). Persistent Personal Names for Globally Connected Mobile Devices. *OSDI '08*.
- Lamport, L. (1998). The Part-Time Parliament. *ACM TOCS*, 16(2), 133–169. (For consensus limits vs. DTN/CRDT design.)

14.3 Attestation, Trust, SBOM, Secure Execution

- IETF RATS WG. (2023). *Remote Attestation Procedures (RATS) Architecture.* RFC 9334.
- IETF. (2023). *Entity Attestation Token (EAT).* RFC 9493.
- Trusted Computing Group (TCG). (2019). *TPM 2.0 Library Specification.*
- Intel. (2020). *Intel® SGX DCAP for Attestation*.
- NTIA → CISA. (2021–2023). *Software Bill of Materials (SBOM) Framing & Minimum Elements.*
- SLSA Steering Committee. (2023). *SLSA: Supply-chain Levels for Software Artifacts* v1.0.

14.4 Safety Engineering & Assurance, Risk, Medical/Industrial

- Kelly, T., & Weaver, R. (2004). The Goal Structuring Notation—A Safety Argument Notation. *DSN Workshop*.
- Leveson, N. (2012). *Engineering a Safer World: Systems Thinking Applied to Safety (STAMP).* MIT Press.
- ISO 26262 (2018). *Road Vehicles—Functional Safety.*
- IEC 61508 (2010/2017). *Functional Safety of E/E/PE Safety-Related Systems.*
- ISO 14971 (2019). *Medical Devices—Application of Risk Management to Medical Devices.*
- FDA, Health Canada, MHRA (2021). *Good Machine Learning Practice for Medical Device Development: Guiding Principles.*
- IEEE Std 1547 (2018). *Interconnection and Interoperability of DER with Associated Electric Power Systems Interfaces.*

14.5 Consent, Privacy, Fairness, Accountability

- Dwork, C., & Roth, A. (2014). *The Algorithmic Foundations of Differential Privacy.* Now. doi:10.1561/0400000042

- Hardt, M., Price, E., & Srebro, N. (2016). Equality of Opportunity in Supervised Learning. *NeurIPS ’16*.
- Selbst, A. D., & Barocas, S. (2018). The Intuitive Appeal of Explainable Machines. *Fordham L. Rev.*, 87(3), 1085–1139.
- Rudin, C. (2019). Stop Explaining Black Box ML for High-Stakes Decisions. *Nature Machine Intelligence*, 1, 206–215. doi:10.1038/s42256-019-0048-x
- Wachter, S., Mittelstadt, B., & Russell, C. (2018). Counterfactual Explanations. *Harv. J. L. & Tech.*, 31(2).
- HIPAA (1996, as amended). *Health Insurance Portability and Accountability Act* and Privacy Rule.
- GDPR (2016/2018). *General Data Protection Regulation*—purpose limitation, data minimization.

14.6 Calibration, OOD, Robustness, Goodhart

- Guo, C., et al. (2017). On Calibration of Modern Neural Networks. *ICML ’17*, 1321–1330.
- Hendrycks, D., & Gimpel, K. (2017). A Baseline for Detecting Misclassified and OOD Examples. *ICLR ’17 (workshop).*
- Liu, W., et al. (2020). Energy-Based OOD Detection. *NeurIPS ’20*.
- Arnaudon, A., et al. (2024). A Survey on OOD Detection for Safety-Critical ML. *ACM Computing Surveys* (early access).
- Manheim, D., & Garrabrant, S. (2018). Categorizing Variants of Goodhart’s Law. *arXiv:1803.04585*.

14.7 Objective Layers, Governance, Standards

- NIST (2023). *AI Risk Management Framework (AI RMF 1.0).* NIST AI 100-1.
- ISO/IEC 42001 (2023). *AI Management System Standard.*
- EU (2024). *Artificial Intelligence Act* (final text, risk tiers and obligations).
- Lessig, L. (1999). *Code and Other Laws of Cyberspace.* Basic Books. (Code-as-law frame.)
- Ostrom, E. (1990). *Governing the Commons.* Cambridge. (Local governance, polycentricity.)

14.8 Microgrids, Resilience, DTN-in-Disasters (Use Cases)

- NREL (2018–2024). *Microgrid Design & Operations* (technical reports on islanded operations).
- Cerf, V., et al. (2007). Delay-Tolerant Networking Architecture. RFC 4838.
- Lindell, R., & Loukaitou-Sideris, A. (2020). *Disaster Management and City Resilience.* Routledge.

14.9 Conatus, Adequacy, and the Mesh (Concept Bridges)

- Bennett, J. (1984). *A Study of Spinoza's Ethics.* Hackett.
- Deleuze, G. (1990). *Expressionism in Philosophy: Spinoza.* Zone.
- Curley, E. (1969). *Spinoza's Metaphysics: An Essay in Interpretation.* Harvard.

14.10 This Series (Primary Source for the Term "Embodied Mesh")

- **Youvan, D. C.** (October 2025). *Transistor → CPU → GPU → Model → Agent → Society → Autonomic Systems → Objective Layer → Co-Designed Matter → Embodied Mesh.* ResearchGate preprint. https://www.researchgate.net/publication/396270471_Transistor_CPU_GPU_Model_Agent_Society_Autonomic_Systems_Objective_Layer_Co-Designed_Matter_Embodied_Mesh